

DELL FLUID CACHE™ FOR SAN



3.58x NUMBER OF
NEW ORDERS
PER MINUTE

POWERED BY
INTEL® XEON®
E5-2660 V3
PROCESSORS

On three Dell™ PowerEdge™ R630 servers running Dell Fluid Cache for SAN and a two-node Oracle® Database 12c Standard Edition RAC



with Express
Flash NVMe
PCIe® SSDs

vs a two-node Oracle Database 12c Standard Edition RAC without Dell Fluid Cache or SSDs



The faster customers and employees can access their data, the better. Using caching software can speed up your applications to handle more transactions and deliver a better experience for end users.

In our hands-on tests in the Principled Technologies labs, we tested two configurations. The first configuration was a two-server Oracle Database 12c Standard Edition Real Application Cluster (RAC), running on two new Intel Xeon processor E5-2660 v3-powered Dell PowerEdge R630 servers. The second configuration used the same two-server Oracle RAC configuration, and added Dell Fluid Cache for SAN to the picture. These additions included the Dell Fluid Cache for SAN software, a third Dell PowerEdge R630 server, and Dell PowerEdge Express Flash NVMe PCIe solid state drives (SSDs). We compared the Oracle Database 12c Standard Edition RAC performance of this Dell Fluid Cache for SAN configuration to that of the original two-server Dell PowerEdge R630 Oracle RAC without Dell Fluid Cache for SAN.

We found that using Dell Fluid Cache for SAN software delivered 3.58 times the Oracle database performance of the two Oracle RAC servers and reduced disk latency, a measure of how long it takes to complete an I/O request, by 91.6 percent.

Investing in Dell Fluid Cache for SAN and Dell PowerEdge Express Flash NVMe PCIe SSDs can deliver greater data acceleration for your database workloads, so that your applications can handle more transactions in the same time.



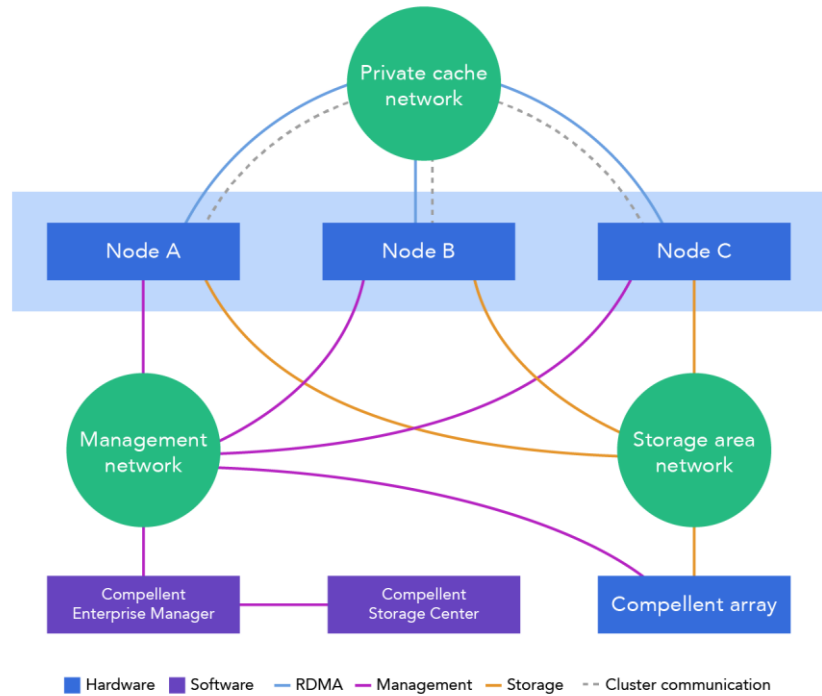
HOW FLUID CACHE FOR SAN SPEEDS UP PERFORMANCE

Caching software solutions duplicate frequently accessed data in the cache layer to reduce application/disk latencies and user response times, by minimizing the amount of data that travels from the slower storage media to the application interface. Caching is ideal for less predictable workloads that may experience spikes in activity, or “hotness” in certain areas, such as database activity during a busy shopping day for a retailer.

Dell designed Fluid Cache for SAN to accelerate applications and boost transactional performance. Fluid Cache for SAN utilizes direct-attached Dell PowerEdge Express Flash NVMe PCIe SSDs and remote direct memory access (RDMA) networking capabilities to create a clustered caching pool for improved storage workload performance. Applications write directly to the cache pool, the acknowledgement is sent back to the application, and in the background the IO is flushed to the external storage array. Data is copied in to the cache pool when it is accessed (read) by application and is available from the cache pool for re-reads. Additionally, data that is written into the cache pool by the application is then available for re-reads from the application. A high-speed link to other cache nodes in the Fluid Cache for SAN pool allows your data to remain highly available by replicating blocks of data, even if a single node in cache pool goes down. Fluid Cache for SAN provides a single management interface and provides other capabilities such as cache-aware snapshots and replication and compression. Dell Fluid Cache allows you to easily create and add the cache you need to make sure you meet the current and future demands of your business.

Figure 1 shows an example environment highlighting how Fluid Cache for SAN communicates between server and storage. Three nodes are required to run Dell Fluid Cache for SAN software. Two of these three nodes are required to have a minimum of one Dell PowerEdge Express Flash NVMe PCIe SSD each, and the 10GbE or 40GbE network for private cache network is based on the low-latency RDMA protocol. The RDMA network handles the cache pool communication between the nodes. Creation and management of the Fluid Cache cluster is accomplished using the Dell Compellent Enterprise Manager software through automatic cache server discovery.

Figure 1: Dell Fluid Cache for SAN communication.

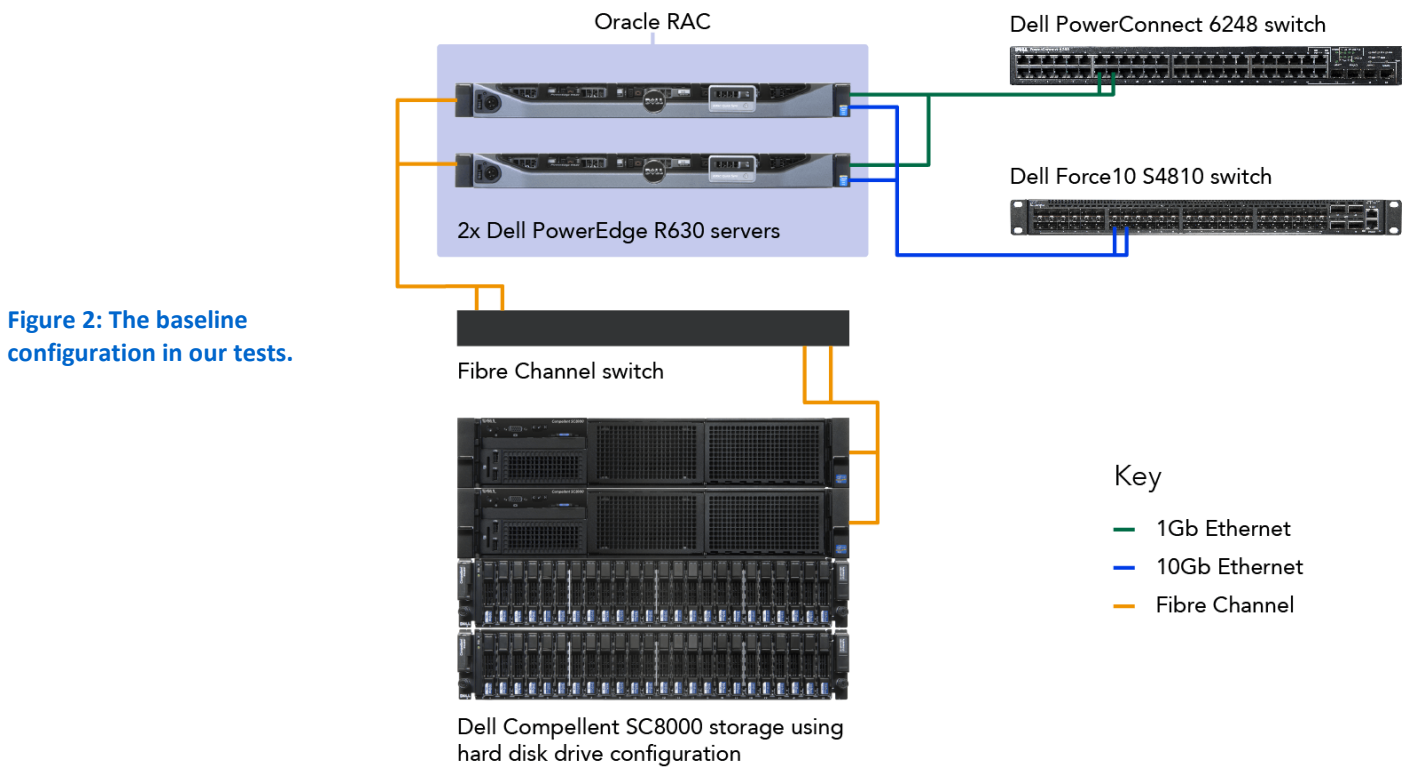


Three nodes are required when running Dell Fluid Cache for SAN software in order to provide resiliency and replicate data in the cache pool. Two nodes must have at least one PCIe SSD each, and the network is a high-speed RDMA network. The RDMA network handles the cluster caching communication between the nodes. Creation of the Fluid Cache cluster on the SAN is accomplished using the Dell Compellent Enterprise Manager software through automatic cache server discovery through the Fluid Cache management network.

OUR TEST CONFIGURATION

We first tested our database workload with two servers running Oracle 12c database Standard Edition in an Oracle RAC configuration (Baseline solution) running on SUSE Linux Enterprise 11 Service Pack 3. In this configuration, only Oracle RAC database software ran on the two nodes. Figure 2 depicts our test setup for the baseline solution.

Two-node Oracle RAC solution



We then added Fluid Cache to the configuration to test the performance improvements. To do so required a third Dell PowerEdge R630. The original two servers continued to run Oracle 12c database Standard Edition RAC and now ran Dell Fluid Cache for SAN software in addition (Dell Fluid Cache for SAN solution). The third server was not part of the Oracle RAC cluster and ran only Dell Fluid Cache for SAN software, though other applications could be installed if necessary. For direct comparison to our first configuration, we did not install additional applications on the third node, and it ran only Dell Fluid Cache for SAN software. (See Figure 3). The entire configuration ran SUSE Linux Enterprise 11 Service Pack 3. We used a 40GbE private cache network.

Dell Fluid Cache solution

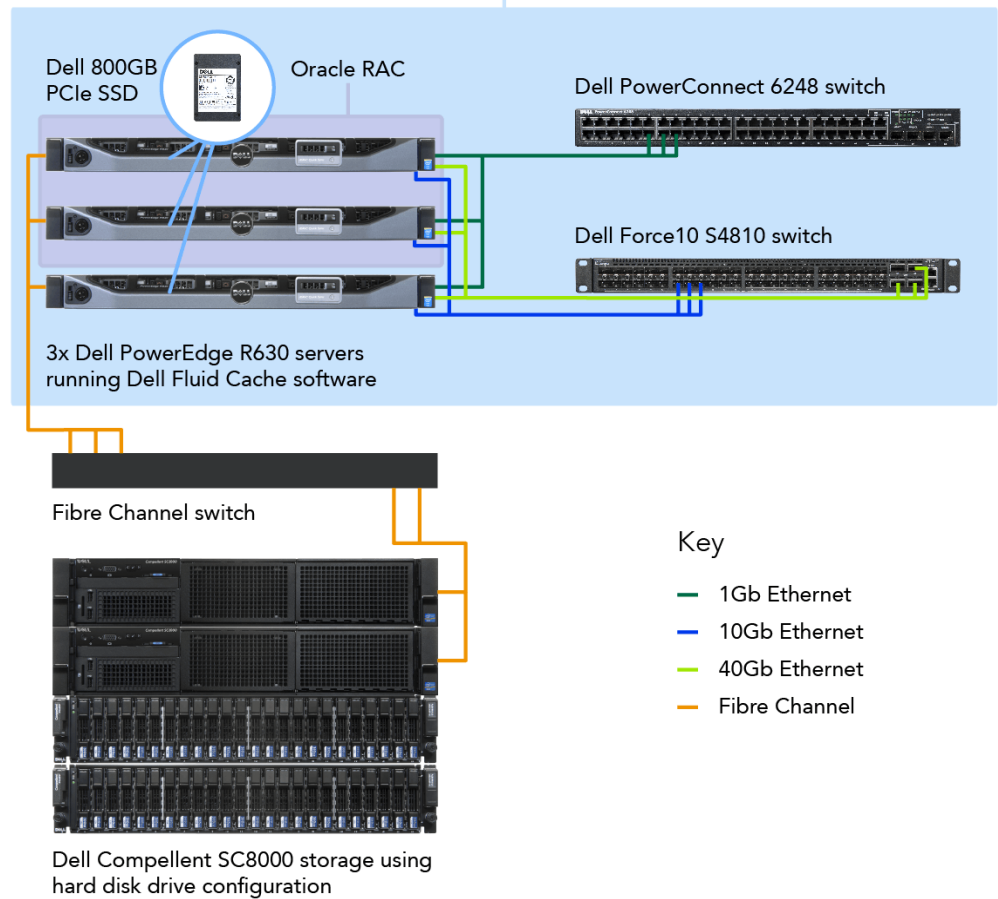


Figure 3: The Dell Fluid Cache for SAN configuration in our tests.

We then used Dell Fluid Cache for SAN to accelerate the workload and measured the difference from the baseline configuration. In each configuration, we used two TPC-C databases, sized approximately each at 420GB, and stored the data on Dell Compellent SC8000 storage.

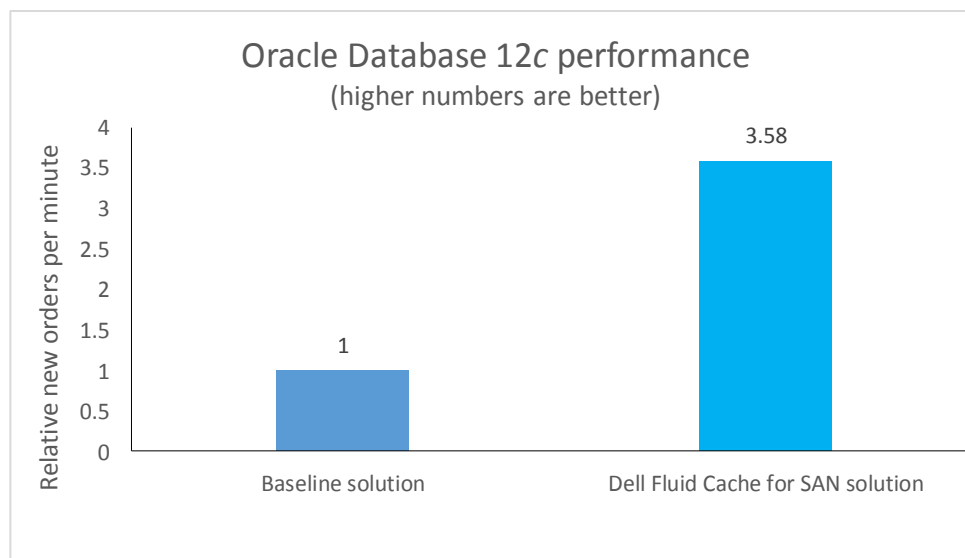
For system configuration information, see [Appendix A](#), and for step-by-step test details, see [Appendix B](#).

WHAT WE FOUND

Dramatic increases in new orders per minute with Dell Fluid Cache for SAN

We found that using Dell Fluid Cache for SAN delivered 3.58 times the Oracle Database 12c Standard Edition performance of our two-node Oracle RAC workload without Fluid Cache for SAN (see Figure 4).

Figure 4: Using Dell Fluid Cache for SAN and Dell PowerEdge Express Flash NVMe PCIe SSDs increased the number of new orders per minute by 3.58x.

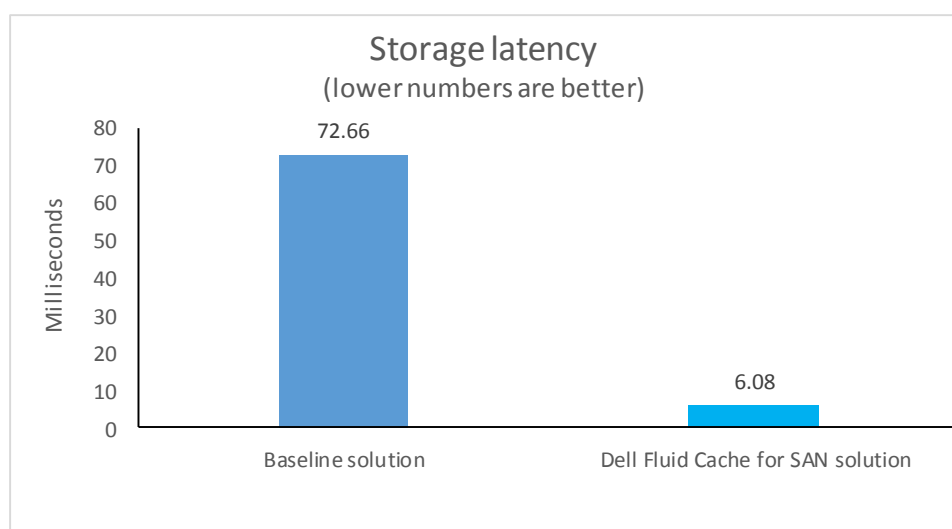


Lower latency with Fluid Cache for SAN means less waiting

Storage latency is a measure of how long it takes to complete an I/O request. Reducing storage latency means reducing the amount of times customers have to wait before their database orders are completed. The more responsive your database's backend storage, the better experience users have and the more business you can do.

We measured the average wait time on our data disks through the Linux tool "sar" over the course of our individual test runs. We report the average "await" (combination of service time and wait time) below. As Figure 5 shows, Dell Fluid Cache for SAN reduced latency on our backend storage by 91.6 percent—down to 6.08 milliseconds compared to the 72.66 milliseconds for the baseline Oracle RAC configuration. The reduction of storage latency, in this case by over 90 percent, plays a huge role in user and application response time, as storage latency is a main driver of user response time.

Figure 5: Using Dell Fluid Cache for SAN and Dell PowerEdge Express Flash NVMe PCIe SSDs decreased storage latency by 91.6 percent



WHAT WE TESTED

About the Dell PowerEdge R630

Fitting into just 1U of rack space, the Dell PowerEdge R630 features two Intel Xeon E5-2600 v3 series processors. Dell designed the PowerEdge R630 with cloud applications, virtualization environments, and high performance computing (HPC) infrastructures in mind. The PowerEdge R630 has 24 DIMM slots to support up to 1.5 TB of memory, supports an optional internal GPU processing accelerator, and can support up to four optional Dell PowerEdge Express Flash NVMe PCIe SSDs.

With redundant power supply units, hot-swappable hardware, and Dual SD card for Failsafe Hypervisors, the Dell PowerEdge R630 supports hardware high availability. The PowerEdge R630 comes standard with iDRAC8 with Lifecycle Controller and Dell OpenManage, which are designed by Dell to streamline management tasks.

About Dell PowerEdge Express Flash NVMe PCIe SSDs

Dell PowerEdge Express Flash NVMe PCIe SSDs are SSDs that run on PCI lanes, which are faster than SAS or SATA interfaces. These types of drives are designed to provide maximum I/O performance and extremely low latencies. Express Flash NVMe PCIe SSDs use nonvolatile NAND flash to replace mechanical drives to avoid IO bottlenecks that traditional hard disk drives can have. Express Flash NVMe PCIe SSDs come in a 2.5" form factor that can be plugged into a device bay.

For more information about PowerEdge Express Flash NVMe PCIe SSDs, visit www.dell.com/learn/us/en/04/campaigns/poweredge-express-flash.

About the benchmark we used - HammerDB

HammerDB is an open-source benchmark tool that tests the database performance of many databases, including Oracle Database, Microsoft® SQL Server®, PostgreSQL®, MySQL™, and more. The benchmark includes built-in workloads derived from industry-standard benchmarks, such as a transactional (TPC-C-like) workload and a data warehouse (TPC-H-like) workload. For this study, we used the transactional workload. Our tests were not official TPC results and are not comparable in any manner. For more information about HammerDB, visit hammerora.sourceforge.net.

CONCLUSION

Don't let your servers hold back your business. Making strides in the performance your infrastructure delivers doesn't mean you need to replace your entire datacenter. With new technology, adding some key pieces can get the performance you need from your critical database workloads.

As we found in our tests, using Dell Fluid Cache for SAN with PowerEdge Express Flash NVMe PCIe SSDs provided 3.58 times the Oracle Database 12c Standard Edition RAC performance of a baseline Oracle RAC cluster. The Dell Fluid Cache for SAN solution

also reduced latency by 91.6 percent. This means that for similar workloads and setups, you could see great benefits from using this quicker storage solution, including quicker database responsiveness and less lag for end users.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 6 provides detailed configuration information for the test systems.

System	3 x Dell PowerEdge 630
Power supplies	
Total number	2
Vendor and model number	Dell E750E-S1
Wattage of each (W)	750
Cooling fans	
Total number	7
Vendor and model number	Delta GFM0412SS
Dimensions (h x w) of each	1.6" x 1.6"
Volts	12
Amps	2.00
General	
Number of processor packages	2
Number of cores per processor	10
Number of hardware threads per core	2
System power management policy	Performance
CPU	
Vendor	Intel
Name	Xeon
Model number	E5-2660 v3
Stepping	2
Socket type	LGA2011-3
Core frequency (GHz)	2.60
Bus frequency	5 GT/s DMI
L1 cache	32 KB + 32 KB (per core)
L2 cache	256 KB (per core)
L3 cache(MB)	25
Platform	
Vendor and model number	Dell PowerEdge R630
Motherboard model number	0CNCJW
BIOS name and version	0.3
BIOS settings	Performance
Memory module(s)	
Total RAM in system (GB)	24
Vendor and model number	Hynix HMA41GR7MFR8N-TF
Type	PC4-2133P
Speed (MHz)	1600
Speed running in the system (MHz)	1600
Timing/Latency (tCL-tRCD-tRP-tRASmin)	11-11-11-33
Size (GB)	8
Number of RAM module(s)	8

System	3 x Dell PowerEdge 630
Chip organization	Double Sided
Rank	2
Operating system	
Name	SUSE Linux Enterprise Server 11 (x86_64)
Build number	11 SP3
File system	ext3
Kernel	3.0.76-0.11-default
Language	English
Graphics	
Vendor and model number	Matrox® G200e
Graphics memory (MB)	8
RAID controller	
Vendor and model number	Dell H730P
Firmware version	25.2.1.0026
Cache size	2 GB
PCIe SSD controller	
Vendor and model number	Dell UCEA200
Type	PCIe
Hard drives	
Vendor and model number	Dell ST300MM0006
Number of drives	2
Size (GB)	300
RPM	10,000
Type	SAS
RAID	RAID 1
Hard drives 2	
Vendor and model number	Dell MZ-WEI8000
Number of drives	1
Size (GB)	800
RPM	SSD
Type	NVMe PCIe SSD
Ethernet adapters	
First network adapter	
Vendor and model number	Mellanox 8KP6W 40G ConnectX-3
Type	PCIe
Vendor and model number	Brocade 7T5GY 8G PCIe Dual Port Fiber
Type	PCIe
Third network adapter	
Vendor and model number	Broadcom® 5720-t 1GbE
Type	Integrated

Figure 6: Configuration information for the test systems.

Figure 7 provides detailed information for the test storage.

Storage array		Dell Compellent SC8000
Number of active storage controllers		2
Firmware revision		6.5.2.12
Storage controller model		CT-SC8000
Tray 1		
Number of disks		24
Disk vendor and model number		Dell ST9146853SS
Disk size (GB)		146
Disk buffer size (MB)		64
Disk RPM		15,000
Disk type		SAS
Tray 2		
Number of disks		24
Disk vendor and model number		Dell ST9146853SS
Disk size (GB)		146
Disk buffer size (MB)		64
Disk RPM		15,000
Disk type		SAS

Figure 7: Storage array configuration information.

APPENDIX B – HOW WE TESTED

Test bed overview

For our baseline testing, we configured two Dell PowerEdge R630 servers in an Oracle RAC cluster running two ~420GB (5000 warehouses)TPC-C databases on Dell Compellent SC8000 storage. Each server contained two HBA cards: a dual port Brocade 8Gb Fibre adapter and a dual-port Mellanox ConnectX-3 40Gb adapter. We added a 40Gb to 10Gb adapter to one port on the Mellanox adapter on each server and cabled it to a 10Gb port on a Dell Force10™ S4810 switch. This network was used for our ASM and Oracle RAC virtual interconnect traffic. We also connected a single onboard 1Gb connection to a Dell PowerConnect™ 6248 switch for our Oracle RAC management traffic and our client traffic.

For our storage network, we connected one Fibre port from each server to a Brocade 8Gb Fibre switch. Our Dell Compellent SC8000 was also attached to the Fibre switch via a single Fibre connection from each SC8000 controller. Our storage layout consisted of the following LUNs:

- 1 x 1.5TB LUN for Backup
- 1 x 30GB LUN for the Oracle RAC Cluster Registry Volume
- 4 x 400GB LUNs for our Oracle ASM Data disk group
- 4 x 40GB LUNs for our Oracle ASM Log disk group

We added our servers to the Dell Compellent Enterprise Manager, targeting each individual Fibre Host Bus Adapter. Each LUN was then mapped to each R630. For a detailed cabling layout, see Figure 2.

For our Fluid Cache testing, we used the previously described configuration with the following changes. Due to Fluid Cache prerequisites, we added a third Dell PowerEdge R630. We also installed three 800GB Dell PowerEdge Express Flash NVMe PCIe SSDs, one per server. For our Fluid Cache network, we cabled the second Mellanox 40Gb port on the Oracle RAC nodes as well as a Mellanox 40Gb port on the third R630 to the 40Gb ports on the Dell Force10 S4810. The third PowerEdge R630 also had a 1GbE management connection to the Dell PowerConnect 6248. We installed the Fluid Cache software on each of the three servers and created the Fluid Cache cluster with Dell Compellent Enterprise Manager. For a detailed cabling layout, see Figure 3.

Software installation and configuration

Installing SuSE Linux Enterprise Server 11 SP3 on the Dell PowerEdge R630 servers

With the latest driver pack installed for OS deployment in Lifecycle Controller, complete the following steps to install SuSE Enterprise Linux 11 SP3.

1. Insert the SLES 11 SP3 install media, and boot into the Lifecycle Controller.
2. Select OS Deployment.
3. Select Deploy OS.
4. On the Select an Operating System screen, select BIOS.
5. In the OS selection drop down, select SuSE Enterprise Linux 11 SP3 x64, and click Next.
6. On the Select Installation Mode, select Manual Install, and click Next.
7. On the Insert OS Media screen, ensure that the install media is attached to the server, and click Next.
8. On the Reboot the System screen, review the OS deployment options, and click Finish.
9. Allow the system to boot into the SuSE welcome menu, and select Installation.
10. At the Welcome screen, agree to the license terms, and click Next.

11. On the Media Check screen, click Next.
12. Wait for the System Probing to complete.
13. On the Installation Mode screen, select New Installation, and click Next.
14. On the Clock and Time Zone screen, select the proper Region and Time Zone, and click Next.
15. On the Server Base Scenario screen, select Physical Machine, and click Next.
16. On the Installation Settings screen, select Expert.
17. Select Partitioning.
18. Select Custom Partitioning (for experts), and click Next.
19. Navigate to Hard Disks→sda.
20. Select the swap partition, and click Delete.
21. At the warning, click Yes.
22. Select the second partition, and click Resize.
23. Select Custom Size, and subtract 20GB.
24. Click OK, then click Add.
25. Select Primary Partition, and click Next.
26. For the New Partition Size, enter 20GB, and click Next.
27. Under Formatting Option, select Swap as the file system and click Finish.
28. Back on the Expert Partitioner screen, click Accept.
29. On the Installation Settings screen, select Kdump.
30. Select Disable Kdump, and click OK.
31. Click Install.
32. Agree to the license, then click Install to begin the installation.
33. Once the install is complete, server will reboot into SLES initial configuration steps.
34. On the Password for the System Administrator “root” screen, enter and confirm a password, and click Next.
35. On the Hostname and Domain Name screen, enter the desired hostname and domain information, and click Next.
36. On the Network Configuration screen, click Disable IPv6.
37. Under Firewall, click disable.
38. Select Network Interfaces.
39. Select the port connected to the private network, and click Edit.
40. Select Statically assigned IP address, and enter the desired information for IP Address and Subnet Mask.
41. Click Next.
42. Click Hostname/DNS, and enter the Name Server IP Address.
43. Click Routing, and enter the desired default gateway IP Address.
44. Click OK.
45. Click VNC Remote Administration, and click Allow Remote Administration.
46. Click Finish, and then click Next.
47. On the Test Internet Connection screen, select Yes, and click Next.
48. When the test completes, click Next.
49. On the Novell Customer Center Configuration screen, click Configure Later, and click Next.
50. On the Network Services Configuration screen, accept defaults, and click Next.
51. On the User Authentication Method screen, select Local, and click Next.
52. On the New Local User screen, click Next.
53. At the Empty User Login prompt, click Yes.
54. On the Release Notes screen, click Next.
55. When prompted, click OK.
56. Once the installer completes the discovery of connection hardware, click Next.
57. On the Installation Completed screen, deselect Clone This System for AutoYaST, and click Finish.

58. Restart the system to ensure that all changes are complete.

Configuring SLES 11 SP3 for Oracle RAC

1. Log onto the server as root.
2. Download and install the Mellanox driver from Dell (version 2.0-2.6.8).
3. Download and install the latest drivers for the Fibre adapter.
4. Start YaST2
5. Click Network Settings
6. Select the Hostname/DNS tab.
7. Enter a hostname, domain name, and name server.
8. Select the Routing tab.
9. Enter a default gateway.
10. Select the Overview tab.
11. Select the first onboard port for client traffic, and click Edit.
12. On the General tab, set the Firewall Zone to Firewall Disabled.
13. On the Address tab, select Statically assigned IP address.
14. Enter an IP address, Subnet Mask, and Hostname; and click Next.
15. Repeat steps 9-12 for the two Mellanox ports for Fluid Cache and ASM traffic.
16. Click OK.
17. Click Firewall.
18. Select Disable Firewall Automatic Starting, click Next, and click Finish.
19. Click NTP Configuration.
20. Click Now and On Boot.
21. Click Add.
22. Select Server, and click Next.
23. Enter the IP address of the NTP server, and click OK.
24. Click OK to complete the configuration wizard.
25. Back on the YaST2 menu, select Software Management.
26. Select the following packages using these versions or later:
 - binutils-2.21.1-0.7.25
 - gcc-4.3-62.198
 - gcc-c++-4.3-62.198
 - glibc-2.11.3-17.31.1
 - glibc-devel-2.11.3-17.31.1
 - ksh-93u-0.6.1
 - libaio-0.3.109-0.1.46
 - libaio-devel-0.3.109-0.1.46
 - libcap1-1.10-6.10
 - libstdc++33-3.3.3-11.9
 - libstdc++33-32bit-3.3.3-11.9
 - libstdc++43-devel-4.3.4_20091019-0.22.17
 - libstdc++46-4.6.1_20110701-0.13.9
 - libgcc46-4.6.1_20110701-0.13.9
 - make-3.81
 - sysstat-8.1.5-7.32.1
 - xorg-x11-libs-32bit-7.4
 - xorg-x11-libs-7.4
 - xorg-x11-libX11-32bit-7.4

- xorg-x11-libX11-7.4
- xorg-x11-libXau-32bit-7.4
- xorg-x11-libXau-7.4
- xorg-x11-libxcb-32bit-7.4
- xorg-x11-libxcb-7.4
- xorg-x11-libXext-32bit-7.4
- xorg-x11-libXext-7.4

27. Click Accept.

28. If a prompt appears, click Continue to install the select packages.

29. Close YaST2.

30. Create Oracle users and groups by running these shell commands:

```
groupadd -g 1001 oinstall
groupadd -g 1002 dba
groupadd -g 1003 asmadmin
groupadd -g 1004 asmdba
useradd -m -u 1002 -g oinstall -G dba,asmadmin,asmdba oracle
useradd -m -u 1003 -g oinstall -G dba,asmadmin,asmdba grid
```

31. Add the following lines to the .profile file for each user:

```
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=(server hostname)
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=$ORACLE_BASE/product/12.1.0/grid
export DB_HOME=$ORACLE_BASE/product/12.1.0/dbhome_1
export ORACLE_HOME=$DB_HOME
export ORACLE_SID=orcl
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
alias grid_env='. /home/oracle/grid_env'
alias db_env='. /home/oracle/db_env'
```

32. Create the following files in each user's home folder.

```
>>> db_env <<<
export ORACLE_SID=orcl
export ORACLE_HOME=$DB_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

```
>>>grid_env<<<
export ORACLE_SID=+ASM
export ORACLE_HOME=$GRID_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

33. Create the following directories, and assign the following permissions.

```
mkdir -p /u01/app/oracle
mkdir /u01/app/grid
chown -R oracle:oinstall /u01/app
chmod -R g+w /u01/app
```

34. Create passwords for the oracle and grid accounts with passwd.

35. Append the following to /etc/security/limits.conf:

```
oracle -   nofile   65536
oracle -   nproc    16384
oracle -   stack    32768
oracle -   memlock  152043520
grid -     nofile   65536
grid -     nproc    16384
grid -     stack    32768
* soft memlock unlimited
* hard memlock unlimited
```

36. We modified the system's kernel parameters by appending the following to /etc/sysctl.conf:

```
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
kernel.shmmni = 4096
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
net.core.rmem_default = 262144
net.core.rmem_max = 20971520
net.core.wmem_default = 262144
net.core.wmem_max = 20971520
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
vm.nr_hugepages = 88064
vm.hugetlb_shm_group = 1001
```

37. Create the following file in /etc/udev/rules.d/, replacing the ID_SERIAL to match the LUN IDs in Compellent Enterprise Manager.

```
>>>99-oracleasmdevices.rules<<<
KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f00000000000000000034", NAME="oracleasm/CRV1",
OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block", PROGRAM=="/lib/udev/scsi_id -g -u -
d $tempnode", RESULT=="36000d31000ee8f0000000000000000002c",
NAME="oracleasm/DATA1", OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block", PROGRAM=="/lib/udev/scsi_id -g -u -
d $tempnode", RESULT=="36000d31000ee8f0000000000000000002d",
NAME="oracleasm/DATA2", OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block", PROGRAM=="/lib/udev/scsi_id -g -u -
d $tempnode", RESULT=="36000d31000ee8f0000000000000000002e",
NAME="oracleasm/DATA3", OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block", PROGRAM=="/lib/udev/scsi_id -g -u -
d $tempnode", RESULT=="36000d31000ee8f0000000000000000002f",
NAME="oracleasm/DATA4", OWNER="grid", GROUP="asmadmin", MODE="0660"
```



```

KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f0000000000000000030", NAME="oracleasm/LOG1",
OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f0000000000000000031", NAME="oracleasm/LOG2",
OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f0000000000000000032", NAME="oracleasm/LOG3",
OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f0000000000000000036", NAME="oracleasm/LOG4",
OWNER="grid", GROUP="asmadmin", MODE="0660"
KERNEL=="sd*|fldc*", SUBSYSTEM=="block",
ENV{ID_SERIAL}=="36000d31000ee8f0000000000000000035",
NAME="oracleasm/BACKUP", OWNER="grid", GROUP="asmadmin", MODE="0660"

```

Installing Oracle Grid and RAC

1. Log into the server as the grid account.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Copy the extracted download files for the Grid install to /grid.
4. Run the Grid Infrastructure GUI installer.
./grid/runInstaller
5. On the Download Software Updates screen, select Skip software updates, and click Next.
6. On the Select Installation Option screen, select Install and Configure Grid Infrastructure for a Cluster, and click Next.
7. On the Select Cluster Type screen, select Configure a Standard cluster, and click Next.
8. On the Select Installation Type screen, select Advanced Installation and click Next.
9. On the Select Product Languages screen, keep the default (English), and click Next.
10. On the Grid Plug and Play Information screen, deselect Configure GNS, enter a Cluster Name and a SCAN Name, and click Next.
11. On the Cluster Node Information screen, click Add.
12. On the Add Cluster Node Information pop-up screen, enter the hostname for node two for the Public hostname and a different hostname Virtual Hostnames, and click OK.
13. Click SSH Connectivity.
14. Enter the OS password for the grid user account, click Setup, then click Next.
15. On the Specify Network Interface Usage screen, set em1 to Public, p2p1 to ASM & Private, and the rest to Do Not Use, and click Next.
16. On the Grid Infrastructure Management Repository Option Screen, select No, and click Next.
17. On the Storage Option Information screen, select Use Oracle Flex ASM for storage, and click Next.
18. On the Create ASM Disk Group screen, click Change Discovery Path.
19. On the Change Discovery Path pop-up screen, enter /dev/oracleasm/, and click OK.
20. Change the Disk Group Name to CRV.
21. Select disks CRV1, and click Next.
22. On the Specify ASM Password, select Use same password for these accounts, and enter the ASMSNMP password (twice), and click Next.
23. On the Failure Isolation Support screen, select Do not use Intelligent Platform Management Interface (IPMI), and click Next.
24. On the Privileged Operating System Groups screen, keep the defaults, and click Next.

25. On the specify Installation location screen, enter `/u01/app/grid` for Oracle Base, `/u01/app/11.2.0/grid` for Software Location, and Click Next.
26. On the Create Inventory screen, enter `/u01/app/oraInventory` for the Inventory Directory, and click Next.
27. On the Root script execution configuration screen, click Next.
28. On the Perform Prerequisite Checks screen, click Fix and Check Again (to install the missing `cvuqdisk` package).
29. On the Execute Fixup Scripts pop-up screen, follow the instructions for running the fix-up script on each node. Click OK when done.
30. Click Next and after the check is finished.
31. On the Summary screen, click Install.
32. When the Execute Configuration scripts pop-up screen appears, follow the instructions for running the scripts, and click OK when finished.
33. Click Close to end the installation.

Creating Oracle ASM disk groups for the databases

1. Log into the system as the grid user.
2. Set the X Window `DISPLAY` variable as appropriate for your configuration.
3. Start the ASM configuration assistant, `asmca`.
4. On the Disk Groups tab, click Create.
5. On the Create Disk Group pop-up screen, enter `DATA` for the Disk Group Name.
6. Select External (None) for Redundancy.
7. Select `/dev/oracleasm/` for the Disk Path.
8. Select the four Data disks.
9. Click Show Advanced Options, and set the ASM and Database compatibilities to 12.1.0.0.
10. Click OK to create the `DATA` disk group, and click OK on the completed-task pop-up screen
11. Repeat steps 4-10 to create disk groups for log and backup.
12. Click Exit to close the ASM configuration assistant.

Installing the software for Oracle Database 12c

1. Log into the system as the oracle user.
2. Set the X Window `DISPLAY` variable as appropriate for your configuration.
3. Copy the extracted download files for the Database install to `/database`.
4. Run the Database GUI installer.
`./database/runInstaller`
5. On the Configure Security Updates screen, unselect I wish to receive security updates via My Oracle Support, and click Next.
6. On the warning pop-up screen, click Yes.
7. On the Download Software Updates screen, select Skip software updates, and click Next.
8. On the Select Installation Option screen, select Install database software only, and click Next.
9. On the Grid Installation Options screen, select Oracle Real Application Clusters database installation, select the nodes in the cluster, and click SSH Connectivity.
10. Enter the OS password for the grid user account, click Setup, then click Next.
11. On the Select Product Languages screen, keep the default (e.g., English) and click Next.
12. On the Select Database Edition screen, select Standard Edition and click Next.
13. On the Specify Installation Location screen, enter `/u01/app/oracle` for Oracle Base and `/u01/app/oracle/product/12.1.0/dbhome_1` for Software Location and click Next.
14. On the Privileged Operating System Groups screen, keep the defaults and click Next.
15. On the Perform Prerequisite Checks screen, click Next.
16. On the Summary screen, click Install.
17. When the Execute Configurations scripts screen appears, follow the instructions and click OK.

18. On the final screen, click Close to end the installation.

Creating the database

We used the following steps to create our Oracle RAC databases. For specific spfile configurations, see [Appendix](#)

[D](#).

1. Log into the system as the oracle user.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Start the database configuration assistant, dbca.
4. On the Database Operations screen, select Create a Database and click Next.
5. On the Creation Mode screen, select Advanced Mode, and click Next.
6. On the Database Templates screen, select Oracle Real Application Clusters (RAC) database for the Database Type, select Admin-Managed for the Configuration Type, and select General Purpose or Transaction Processing.
7. Click Next.
8. On the Database Identification screen, enter `orcl` for Global Database Name and SID Prefix, and click Next.
9. On the Database Placement screen, add both RAC nodes, and click Next.
10. On the Management Options screen, keep the defaults and click Next.
11. On the Database Credentials screen, select Use the Same Administrative Password for All Accounts, enter the password twice, and click Next.
12. On the Storage Locations screen, select ASM for Storage Type, select Use Common Location for All Database Files, enter `+DATA` for Database Files Locations.
13. Under Recovery Related Files, select ASM for the Storage Type, enter `+BACKUP` for the Fast Recovery Area, and enter 1.5 TB for the Fast Recovery Size.
14. On the Initialization Parameters screen, leave the defaults, and click Next.
15. On the Creation Options screen, select Create Database, and click Next.
16. On the Pre Requisite Checks screen, allow the prerequisite checker to complete, resolve any issues, and click Next.
17. On the Summary pop-up screen, click OK.
18. Click Exit on the final pop-up screen to close the configuration assistant.
19. Repeat steps 3-18 for the second RAC database.
20. Once the databases installed, we created two redo log groups per database per node. Each contained a single 4GB file on the `+LOG` ASM group. All default groups were removed.
21. On each database, we created a tablespace named TPCC, set it to bigfile at 600GB, and enabled autoextend at 1GB intervals.

Generate HammerDB data

We generated the data using two Oracle Enterprise Linux clients (one per database) with Oracle Database 12c Client and HammerDB installed.

1. Download and install the Oracle Database 12c client from www.oracle.com.
2. Download and install HammerDB from hammerora.sourceforge.net/download.html
3. Edit the `tnsnames` file located in the Oracle home network admin folder to target the virtual IP and SID for the first node and database.
4. Repeat the edit on the second client, targeting the virtual IP and SID for the second node and database.
5. Launch the HammerDB UI.
6. In the HammerDB UI, click Options→Benchmark, and check Oracle and TPC-C. Click OK.
7. Click OK again to confirm the benchmark choice.
8. Expand TPC-C and Schema Build.
9. Double-click Options to open the Build Options menu.
10. For Oracle Service Name, type the SID of the first database: `orcl`.

11. Change the system user password to match the password set during the database installation.
12. Enter TPCC for the default tablespace.
13. Set the number of warehouses to 5,000, and the users to 16.
14. Leave the rest of the fields as default.
15. To start the database generation, double-click Build.

Configuring Fluid Cache (Cache testing only)

We used the following steps to configure and enable Fluid Cache for our caching tests. Before configuring Fluid Cache, we assigned a static IP in a new subnet on the 40Gb ports to be used for the Fluid Cache network.

Installing Fluid Cache

1. Prior to installing Fluid Cache, add the following lines to /etc/multipath.conf on each server:


```
blacklist {
    devnode "^fldc[0-9] *"
    devnode "^nvme[a-z] *"
    devnode "^rssh[a-z] *"
}
```
1. Copy and extract the Fluid Cache software on each server.
2. In the extracted directory, run the rpm -i command on the Fluid Cache RPM to install.
3. Navigate to the /opt/dell/fluidcache/bin directory, and run ./hcn_setup.py.
4. Press Enter to begin the configuration.
5. Enter the desired subnet for the management interface (i.e. 192.168.1.0). This was our 1Gb management and client subnet.
6. Enter the desired subnet mask for the connection.
7. Enter Y to confirm the settings.
8. Enter N to skip configuring another network.
9. Enter the desired subnet for the cache network (i.e. 192.168.2.0). This was our 40Gb network for Fluid Cache.
10. Enter the desired subnet mask for the connection.
11. Enter Y to confirm the settings.
12. Enter Y to start the Fluid Cache software.

Configuring the Fluid Cache cluster

1. On a separate machine running Dell Compellent Enterprise Manager connected to the client network, login in to the EM.
2. At the top on the page, click Configure Fluid Cache Cluster.
3. Enter the management IP of one of the three R630 servers, and enter the root user and password.
4. Click Next.
5. Ensure that all servers are discovered and selected, and click Next.
6. Type a name for the new cluster and select a license file for the new cluster.
7. Click Next.
8. On the Select Devices page, select all three 800GB Dell PowerEdge Express Flash NVMe PCIe SSDs, and click Next.
9. Select the desired Storage Center for the Fluid Cache Cluster that contains the Oracle LUNs, and click Finish.

To enable Fluid Cache on the Data LUNs, unmap the four LUNs from the Oracle RAC nodes. Then remap the LUNs to the Oracle nodes, checking the box to enable Fluid Cache before click finish. In this testing, we used the default Fluid Cache settings for each data LUN.

Running the test

HammerDB settings

We configured HammerDB to run with the AWR snapshot script for a 30 minute warmup time and a 60 minute run. To ensure we had sufficient load on the environment, we set the user count 101 (one user is used as a management user) and left the HammerDB defaults for think time (0). We also selected the options to Show Output and Log Output to Temp. To receive real-time data during the run, we configured the Transaction Counter to monitor TPM performance. Once the run completed, we gathered the NOPM output used this number for our comparison. Due to the nature of multi-node RAC and HammerDB output gathering, we had to edit the run script to gather NOPM and TPM results from both servers and combine them. The driver script we used is located in [Appendix E](#) with the customization in **bold**.

Run workflow

We used the following steps during each run iteration. After generating the two databases, we used RMAN to create a backup of each. These backups were stored in our Fast Recovery area on +BACKUP.

1. Restore the database using RMAN.
2. Reboot the server.
3. Once the two nodes are backup, allow them to sit idle for 20 minutes to ensure that Oracle RAC startup processes have completed.
4. Ensure that all setting in the HammerDB UI for each client are configured properly.
5. Start the Transaction Counter, and click Create Users
6. Start server stats gathering on each RAC node. We used sar to output system statistics to a bin file for 95 minutes using the following script:
`nohup sar -o /tmp/sar_$(hostname -s).bin 15 380 > /dev/null 2>&1 &`
7. On the HammerDB UI, click the green → at the top of the page to begin the test.
8. Once the test completes, gather the sar bin files from the RAC nodes and the HammerDB output log on each client.

APPENDIX D –ORACLE SPFILES

Database: ORCL

```
orcl1.__data_transfer_cache_size=0
orcl2.__data_transfer_cache_size=0
orcl1.__db_cache_size=52881784832
orcl2.__db_cache_size=52747567104
orcl1.__java_pool_size=939524096
orcl2.__java_pool_size=939524096
orcl1.__large_pool_size=1342177280
orcl2.__large_pool_size=1342177280
orcl1.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl2.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl1.__pga_aggregate_target=20401094656
orcl2.__pga_aggregate_target=20401094656
orcl1.__sga_target=60934848512
orcl2.__sga_target=60934848512
orcl1.__shared_io_pool_size=402653184
orcl2.__shared_io_pool_size=536870912
orcl1.__shared_pool_size=4966055936
orcl2.__shared_pool_size=4966055936
orcl1.__streams_pool_size=0
orcl2.__streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/orcl/adump'
*.audit_trail='db'
*.cluster_database=true
*.compatible='12.1.0.0.0'
*.control_files='+DATA/ORCL/CONTROLFILE/current.261.855053573','+BACKUP/ORCL/CONTROLFILE/current.256.855053573'#Restore Controlfile
orcl1.db_16k_cache_size=0
orcl2.db_16k_cache_size=0
*.db_block_size=8192
orcl1.db_cache_size=0
orcl2.db_cache_size=0
*.db_create_file_dest='+DATA'
*.db_domain=''
*.db_name='orcl'
*.db_recovery_file_dest='+BACKUP'
*.db_recovery_file_dest_size=1500g
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB) '
*.filesystemio_options='setall'
orcl1.instance_number=1
orcl2.instance_number=2
*.log_checkpoints_to_alert=FALSE
*.open_cursors=300
*.pga_aggregate_target=19370m
*.processes=300
*.remote_login_passwordfile='exclusive'
*.sga_target=58110m
orcl2.thread=2
```

```
orcl1.thread=1
*.trace_enabled=FALSE
orcl2.undo_tablespace='UNDOTBS2'
orcl1.undo_tablespace='UNDOTBS1'
*.use_large_pages='ONLY'
```

Database: NELS

```
nels2.__data_transfer_cache_size=0
nels1.__data_transfer_cache_size=0
nels2.__db_cache_size=52747567104
nels1.__db_cache_size=52076478464
nels2.__java_pool_size=939524096
nels1.__java_pool_size=939524096
nels2.__large_pool_size=1342177280
nels1.__large_pool_size=1342177280
nels1.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
nels2.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
nels2.__pga_aggregate_target=20401094656
nels1.__pga_aggregate_target=20401094656
nels2.__sga_target=60934848512
nels1.__sga_target=60934848512
nels2.__shared_io_pool_size=402653184
nels1.__shared_io_pool_size=402653184
nels2.__shared_pool_size=5100273664
nels1.__shared_pool_size=5771362304
nels2.__streams_pool_size=0
nels1.__streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/nels/adump'
*.audit_trail='db'
*.cluster_database=true
*.compatible='12.1.0.0.0'
*.control_files='+DATA/NELS/CONTROLFILE/current.279.855064027','+BACKUP/NELS/CONTROLFILE/current.261.855064029'#Restore Controlfile
nels2.db_16k_cache_size=0
nels1.db_16k_cache_size=0
*.db_block_size=8192
nels2.db_cache_size=0
*.db_create_file_dest='+DATA'
*.db_domain=''
*.db_name='nels'
*.db_recovery_file_dest='+BACKUP'
*.db_recovery_file_dest_size=1500g
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=nelsXDB)'
*.filesystemio_options='setall'
nels2.instance_number=2
nels1.instance_number=1
*.log_checkpoints_to_alert=FALSE
*.open_cursors=300
*.pga_aggregate_target=19370m
*.processes=300
```

```
*.remote_login_passwordfile='exclusive'  
*.sga_target=58110m  
nls2.thread=2  
nls1.thread=1  
*.trace_enabled=FALSE  
nls1.undo_tablespace='UNDOTBS1'  
nls2.undo_tablespace='UNDOTBS2'  
*.use_large_pages='ONLY'
```


APPENDIX E: HAMMERDB DRIVER SCRIPT

Below is the driver script we used in HammerDB. Customizations are in **bold**.

```
#!/usr/local/bin/tclsh8.6
if [catch {package require Oratcl} ] { error "Failed to load Oratcl - Oracle OCI
Library Error" }
#AWR SNAPSHOT DRIVER SCRIPT#####
#THIS SCRIPT TO BE RUN WITH VIRTUAL USER OUTPUT ENABLED
#EDITABLE OPTIONS#####
set total_iterations 10000000 ;# Number of transactions before logging off
set RAISEERROR "false" ;# Exit script on Oracle error (true or false)
set KEYANDTHINK "false" ;# Time for user thinking and keying (true or false)
set CHECKPOINT "true" ;# Perform Oracle checkpoint when complete (true or false)
set rampup 30; # Rampup time in minutes before first snapshot is taken
set duration 60; # Duration in minutes before second AWR snapshot is taken
set mode "Local" ;# HammerDB operational mode
set timesten "false" ;# Database is TimesTen
set systemconnect system/Password1@192.168.1.204/orcl ;# Oracle connect string
for system user
set connect tpcc/tpcc@192.168.1.204/orcl ;# Oracle connect string for tpc-c user
#EDITABLE OPTIONS#####
#CHECK THREAD STATUS
proc chk_thread {} {
    set chk [package provide Thread]
    if {[string length $chk]} {
        return "TRUE"
    } else {
        return "FALSE"
    }
}
#STANDARD SQL
proc standsql { curn sql } {
    set ftch ""
    if {[catch {orasql $curn $sql} message]} {
        error "SQL statement failed: $sql : $message"
    } else {
        orafetch $curn -datavariabile output
        while { [ oramsg $curn ] == 0 } {
            lappend ftch $output
            orafetch $curn -datavariabile output
        }
        return $ftch
    }
}
#Default NLS
proc SetNLS { lda } {
    set curn_nls [oraopen $lda ]
    set nls(1) "alter session set NLS_LANGUAGE = AMERICAN"
    set nls(2) "alter session set NLS_TERRITORY = AMERICA"
    for { set i 1 } { $i <= 2 } { incr i } {
        if {[ catch {orasql $curn_nls $nls($i)} message ] } {
```

```

puts "$message $nls($i)"
puts [ orams $curn_nls all ]
    }
}
oraclose $curn_nls
}

if { [ chk_thread ] eq "FALSE" } {
error "AWR Snapshot Script must be run in Thread Enabled Interpreter"
}
set mythread [thread::id]
set allthreads [split [thread::names]]
set totalvirtualusers [expr [llength $allthreads] - 1]
set myposition [expr $totalvirtualusers - [lsearch -exact $allthreads $mythread]]
if {[catch {set timeout [tsv::get application timeout]}]} {
if { $timeout eq 0 } {
set totalvirtualusers [ expr $totalvirtualusers - 1 ]
set myposition [ expr $myposition - 1 ]
    }
}
if { [ string toupper $timesten ] eq "TRUE" } {
set timesten 1
set systemconnect $connect
} else {
set timesten 0
}
switch $myposition {
1 {
if { $mode eq "Local" || $mode eq "Master" } {
set lda [oralogon $systemconnect]
if { !$timesten } { SetNLS $lda }
set lda1 [oralogon $connect]
if { !$timesten } { SetNLS $lda1 }
oraautocom $lda on
oraautocom $lda1 on
set curn1 [oraopen $lda ]
set curn2 [oraopen $lda1 ]
if { $timesten } {
puts "For TimesTen use external ttStats utility for performance reports"
set sql1 "select (xact_commits + xact_rollbacks) from sys.monitor"
    } else {
set sql1 "BEGIN dbms_workload_repository.create_snapshot(); END;"
oraparse $curn1 $sql1
    }
set ramptime 0
puts "Beginning rampup time of $rampup minutes"
set rampup [ expr $rampup*60000 ]
while { $ramptime != $rampup } {
if { [ tsv::get application abort ] } { break } else { after 6000 }
set ramptime [ expr $ramptime+6000 ]
if { ![ expr { $ramptime % 60000 } ] } {
puts "Rampup [ expr $ramptime / 60000 ] minutes complete ..."

```

```

    }
}
if { [ tsv::get application abort ] } { break }
if { $timesten } {
puts "Rampup complete, Taking start Transaction Count."
set start_trans [ standsql $curn2 $sql1 ]
    } else {
puts "Rampup complete, Taking start AWR snapshot."
if {[catch {oraplexec $curn1 $sql1} message]} { error "Failed to create snapshot
: $message" }
set sql2 "SELECT INSTANCE_NUMBER, INSTANCE_NAME, DB_NAME, DBID, SNAP_ID,
TO_CHAR(END_INTERVAL_TIME,'DD MON YYYY HH24:MI') FROM (SELECT DI.INSTANCE_NUMBER,
DI.INSTANCE_NAME, DI.DB_NAME, DI.DBID, DS.SNAP_ID, DS.END_INTERVAL_TIME FROM
DBA_HIST_SNAPSHOT DS, DBA_HIST_DATABASE_INSTANCE DI WHERE DS.DBID=DI.DBID AND
DS.INSTANCE_NUMBER=DI.INSTANCE_NUMBER AND DS.STARTUP_TIME=DI.STARTUP_TIME ORDER
BY DS.SNAP_ID DESC) WHERE ROWNUM=1"
if {[catch {orasql $curn1 $sql2} message]} {
error "SQL statement failed: $sql2 : $message"
} else {
orafetch $curn1 -datavariabile firstsnap
split $firstsnap " "
puts "Start Snapshot [ lindex $firstsnap 4 ] taken at [ lindex $firstsnap 5 ] of
instance [ lindex $firstsnap 1 ] ([lindex $firstsnap 0]) of database [ lindex
$firstsnap 2 ] ([lindex $firstsnap 3])"
}}
set sql4 "select sum(d_next_o_id) from district"
set start_nopm [ standsql $curn2 $sql4 ]
puts "Timing test period of $duration in minutes"
set testtime 0
set durmin $duration
set duration [ expr $duration*60000 ]
while {$testtime != $duration} {
if { [ tsv::get application abort ] } { break } else { after 6000 }
set testtime [ expr $testtime+6000 ]
if { ![ expr {$testtime % 60000} ] } {
puts -nonewline "[ expr $testtime / 60000 ]    ...,"
    }
}
if { [ tsv::get application abort ] } { break }
if { $timesten } {
puts "Test complete, Taking end Transaction Count."
set end_trans [ standsql $curn2 $sql1 ]
set end_nopm [ standsql $curn2 $sql4 ]
set tpm [ expr {($end_trans - $start_trans)/$durmin} ]
set nopm [ expr {($end_nopm - $start_nopm)/$durmin} ]
puts "$totalvirtualusers Virtual Users configured"
puts "TEST RESULT : System achieved $tpm TimesTen TPM at $nopm NOPM"
    } else {
puts "Test complete, Taking end AWR snapshot."
oraparse $curn1 $sql1
if {[catch {oraplexec $curn1 $sql1} message]} { error "Failed to create snapshot
: $message" }

```

```

if {[catch {orasql $curn1 $sql2} message]} {
error "SQL statement failed: $sql2 : $message"
} else {
orafetch $curn1 -datavARIABLE endsnap
split $endsnap " "
puts "End Snapshot [ lindex $endsnap 4 ] taken at [ lindex $endsnap 5 ] of
instance [ lindex $endsnap 1 ] ([lindex $endsnap 0]) of database [ lindex
$endsnap 2 ] ([lindex $endsnap 3])"
puts "Test complete: view report from SNAPID [ lindex $firstsnap 4 ] to [ lindex
$endsnap 4 ]"
set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value -
b.value) / (select avg( extract( day from (e1.end_interval_time-
b1.end_interval_time) ) *24*60*60+ extract( hour from (e1.end_interval_time-
b1.end_interval_time) ) *60*60+ extract( minute from (e1.end_interval_time-
b1.end_interval_time) ) *60+ extract( second from (e1.end_interval_time-
b1.end_interval_time)) ) from dba_hist_snapshot b1, dba_hist_snapshot e1 where
b1.snap_id = [ lindex $firstsnap 4 ] and e1.snap_id = [ lindex $endsnap 4 ] and
b1.dbid = [lindex $firstsnap 3] and e1.dbid = [lindex $endsnap 3] and
b1.instance_number = 1 and e1.instance_number = 1 and b1.startup_time =
e1.startup_time and b1.end_interval_time < e1.end_interval_time) as tps from
dba_hist_sysstat b, dba_hist_sysstat e where b.snap_id = [ lindex $firstsnap 4 ]
and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and
e.dbid = [lindex $endsnap 3] and b.instance_number = 1 and e.instance_number = 1
and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks')
and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
set tpm1 [ standsql $curn1 $sql3 ]
set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value -
b.value) / (select avg( extract( day from (e1.end_interval_time-
b1.end_interval_time) ) *24*60*60+ extract( hour from (e1.end_interval_time-
b1.end_interval_time) ) *60*60+ extract( minute from (e1.end_interval_time-
b1.end_interval_time) ) *60+ extract( second from (e1.end_interval_time-
b1.end_interval_time)) ) from dba_hist_snapshot b1, dba_hist_snapshot e1 where
b1.snap_id = [ lindex $firstsnap 4 ] and e1.snap_id = [ lindex $endsnap 4 ] and
b1.dbid = [lindex $firstsnap 3] and e1.dbid = [lindex $endsnap 3] and
b1.instance_number = 2 and e1.instance_number = 2 and b1.startup_time =
e1.startup_time and b1.end_interval_time < e1.end_interval_time) as tps from
dba_hist_sysstat b, dba_hist_sysstat e where b.snap_id = [ lindex $firstsnap 4 ]
and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and
e.dbid = [lindex $endsnap 3] and b.instance_number = 2 and e.instance_number = 2
and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks')
and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
set tpm2 [ standsql $curn1 $sql3 ]
set tpm [ expr $tpm1 + $tpm2 ]
set end_nopm [ standsql $curn2 $sql4 ]
set nopm [ expr {($end_nopm - $start_nopm)/$durmin} ]
puts "$totalvirtualusers Virtual Users configured"
puts "TEST RESULT : System achieved $tpm Oracle TPM at $nopm NOPM"
}
}
tsv::set application abort 1
if { $CHECKPOINT } {
puts "Checkpoint"

```

```

if { $timesten } {
set sql4 "call ttCkptBlocking"
    }else {
set sql4 "alter system checkpoint"
if {[catch {oracsql $curn1 $sql4} message]} {
error "SQL statement failed: $sql4 : $message"
}
set sql5 "alter system switch logfile"
if {[catch {oracsql $curn1 $sql5} message]} {
error "SQL statement failed: $sql5 : $message"
}}
puts "Checkpoint Complete"
    }
oraclose $curn1
oraclose $curn2
oralogoff $lda
oralogoff $lda1
    } else {
puts "Operating in Slave Mode, No Snapshots taken..."
    }
}

default {
#RANDOM NUMBER
proc RandomNumber {m M} {return [expr {int($m+rand()*($M+1-$m))}]}
#NURand function
proc NURand { iConst x y C } {return [ expr {(((RandomNumber 0 $iConst) |
(RandomNumber $x $y)) + $C) % ($y - $x + 1)) + $x }]}
#RANDOM NAME
proc randname { num } {
array set namearr { 0 BAR 1 OUGHT 2 ABLE 3 PRI 4 PRES 5 ESE 6 ANTI 7 CALLY 8
ATION 9 EING }
set name [ concat $namearr([ expr {( $num / 100 ) % 10 }])$namearr([ expr {( $num
/ 10 ) % 10 }])$namearr([ expr {( $num / 1 ) % 10 }]) ]
return $name
}
#TIMESTAMP
proc gettimestamp { } {
set tstamp [ clock format [ clock seconds ] -format %Y%m%d%H%M%S ]
return $tstamp
}
#KEYING TIME
proc keytime { keying } {
after [ expr {$keying * 1000} ]
return
}
#THINK TIME
proc thinktime { thinking } {
set thinkingtime [ expr {abs(round(log(rand())) * $thinking))} ]
after [ expr {$thinkingtime * 1000} ]
return
}
#NEW ORDER

```

```

proc neword { curn_no no_w_id w_id_input RAISEERROR } {
#2.4.1.2 select district id randomly from home warehouse where d_w_id = d_id
set no_d_id [ RandomNumber 1 10 ]
#2.4.1.2 Customer id randomly selected where c_d_id = d_id and c_w_id = w_id
set no_c_id [ RandomNumber 1 3000 ]
#2.4.1.3 Items in the order randomly selected from 5 to 15
set ol_cnt [ RandomNumber 5 15 ]
#2.4.1.6 order entry date O_ENTRY_D generated by SUT
set date [ gettimestamp ]
orabind $curn_no :no_w_id $no_w_id :no_max_w_id $w_id_input :no_d_id $no_d_id
:no_c_id $no_c_id :no_o_ol_cnt $ol_cnt :no_c_discount {} :no_c_last {}
:no_c_credit {} :no_d_tax {} :no_w_tax {} :no_d_next_o_id {0} :timestamp $date
if {[catch {oraexec $curn_no} message]} {
if { $RAISEERROR } {
error "New Order : $message [ orams $curn_no all ]"
} else {
;
} } else {
orafetch $curn_no -datavariabile output
;
}
}
#PAYMENT
proc payment { curn_py p_w_id w_id_input RAISEERROR } {
#2.5.1.1 The home warehouse id remains the same for each terminal
#2.5.1.1 select district id randomly from home warehouse where d_w_id = d_id
set p_d_id [ RandomNumber 1 10 ]
#2.5.1.2 customer selected 60% of time by name and 40% of time by number
set x [ RandomNumber 1 100 ]
set y [ RandomNumber 1 100 ]
if { $x <= 85 } {
set p_c_d_id $p_d_id
set p_c_w_id $p_w_id
} else {
#use a remote warehouse
set p_c_d_id [ RandomNumber 1 10 ]
set p_c_w_id [ RandomNumber 1 $w_id_input ]
while { ($p_c_w_id == $p_w_id) && ($w_id_input != 1) } {
set p_c_w_id [ RandomNumber 1 $w_id_input ]
}
}
set nrnd [ NURand 255 0 999 123 ]
set name [ randname $nrnd ]
set p_c_id [ RandomNumber 1 3000 ]
if { $y <= 60 } {
#use customer name
#C_LAST is generated
set byname 1
} else {
#use customer number
set byname 0
set name {}
}
}

```

```

}
#2.5.1.3 random amount from 1 to 5000
set p_h_amount [ RandomNumber 1 5000 ]
#2.5.1.4 date selected from SUT
set h_date [ gettimestamp ]
#2.5.2.1 Payment Transaction
#change following to correct values
orabind $curn_py :p_w_id $p_w_id :p_d_id $p_d_id :p_c_w_id $p_c_w_id :p_c_d_id
$p_c_d_id :p_c_id $p_c_id :byname $byname :p_h_amount $p_h_amount :p_c_last $name
:p_w_street_1 {} :p_w_street_2 {} :p_w_city {} :p_w_state {} :p_w_zip {}
:p_d_street_1 {} :p_d_street_2 {} :p_d_city {} :p_d_state {} :p_d_zip {}
:p_c_first {} :p_c_middle {} :p_c_street_1 {} :p_c_street_2 {} :p_c_city {}
:p_c_state {} :p_c_zip {} :p_c_phone {} :p_c_since {} :p_c_credit {0}
:p_c_credit_lim {} :p_c_discount {} :p_c_balance {0} :p_c_data {} :timestamp
$h_date
if {[ catch {oraexec $curn_py} message]} {
if { $RAISEERROR } {
error "Payment : $message [ orams $curn_py all ]"
} else {
;
} } else {
orafetch $curn_py -datavariabile output
;
}
}
#ORDER_STATUS
proc ostat { curn_os w_id RAISEERROR } {
#2.5.1.1 select district id randomly from home warehouse where d_w_id = d_id
set d_id [ RandomNumber 1 10 ]
set nrnd [ NURand 255 0 999 123 ]
set name [ randname $nrnd ]
set c_id [ RandomNumber 1 3000 ]
set y [ RandomNumber 1 100 ]
if { $y <= 60 } {
set byname 1
} else {
set byname 0
set name {}
}
orabind $curn_os :os_w_id $w_id :os_d_id $d_id :os_c_id $c_id :byname $byname
:os_c_last $name :os_c_first {} :os_c_middle {} :os_c_balance {0} :os_o_id {}
:os_entdate {} :os_o_carrier_id {}
if {[catch {oraexec $curn_os} message]} {
if { $RAISEERROR } {
error "Order Status : $message [ orams $curn_os all ]"
} else {
;
} } else {
orafetch $curn_os -datavariabile output
;
}
}
}

```

```

#DELIVERY
proc delivery { curn_dl w_id RAISEERROR } {
  set carrier_id [ RandomNumber 1 10 ]
  set date [ gettimestamp ]
  orabind $curn_dl :d_w_id $w_id :d_o_carrier_id $carrier_id :timestamp $date
  if {[ catch {oraexec $curn_dl} message ]} {
    if { $RAISEERROR } {
      error "Delivery : $message [ orams $curn_dl all ]"
    } else {
      ;
    } } else {
      orafetch $curn_dl -datavariabile output
      ;
    }
  }
}

#STOCK LEVEL
proc sleve { curn_sl w_id stock_level_d_id RAISEERROR } {
  set threshold [ RandomNumber 10 20 ]
  orabind $curn_sl :st_w_id $w_id :st_d_id $stock_level_d_id :THRESHOLD $threshold
  if {[catch {oraexec $curn_sl} message]} {
    if { $RAISEERROR } {
      error "Stock Level : $message [ orams $curn_sl all ]"
    } else {
      ;
    } } else {
      orafetch $curn_sl -datavariabile output
      ;
    }
  }
}

proc prep_statement { lda curn_st } {
  switch $curn_st {
    curn_sl {
      set curn_sl [oraopen $lda ]
      set sql_sl "BEGIN sleve(:st_w_id,:st_d_id,:threshold); END;"
      oraparse $curn_sl $sql_sl
      return $curn_sl
    }
    curn_dl {
      set curn_dl [oraopen $lda ]
      set sql_dl "BEGIN
delivery(:d_w_id,:d_o_carrier_id,TO_DATE(:timestamp,'YYYYMMDDHH24MISS')); END;"
      oraparse $curn_dl $sql_dl
      return $curn_dl
    }
    curn_os {
      set curn_os [oraopen $lda ]
      set sql_os "BEGIN
ostat(:os_w_id,:os_d_id,:os_c_id,:byname,:os_c_last,:os_c_first,:os_c_middle,:os_
c_balance,:os_o_id,:os_entdate,:os_o_carrier_id); END;"
      oraparse $curn_os $sql_os
      return $curn_os
    }
  }
}

```



```

    }
    curn_py {
    set curn_py [oraopen $lda ]
    set sql_py "BEGIN
payment(:p_w_id,:p_d_id,:p_c_w_id,:p_c_d_id,:p_c_id,:byname,:p_h_amount,:p_c_last
,:p_w_street_1,:p_w_street_2,:p_w_city,:p_w_state,:p_w_zip,:p_d_street_1,:p_d_str
eet_2,:p_d_city,:p_d_state,:p_d_zip,:p_c_first,:p_c_middle,:p_c_street_1,:p_c_str
eet_2,:p_c_city,:p_c_state,:p_c_zip,:p_c_phone,:p_c_since,:p_c_credit,:p_c_credit
_lim,:p_c_discount,:p_c_balance,:p_c_data,TO_DATE(:timestamp,'YYYYMMDDHH24MISS'))
; END;"
oraparse $curn_py $sql_py
return $curn_py
    }
    curn_no {
    set curn_no [oraopen $lda ]
    set sql_no "begin
neword(:no_w_id,:no_max_w_id,:no_d_id,:no_c_id,:no_o_ol_cnt,:no_c_discount,:no_c_
last,:no_c_credit,:no_d_tax,:no_w_tax,:no_d_next_o_id,TO_DATE(:timestamp,'YYYYMMD
DHH24MISS'))); END;"
oraparse $curn_no $sql_no
return $curn_no
    }
}

#RUN TPC-C
set lda [oralogon $connect]
if { !$timesten } { SetNLS $lda }
oraautocom $lda on
foreach curn_st {curn_no curn_py curn_dl curn_sl curn_os} { set $curn_st [
prep_statement $lda $curn_st ] }
set curn1 [oraopen $lda ]
set sql1 "select max(w_id) from warehouse"
set w_id_input [ standsql $curn1 $sql1 ]
#2.4.1.1 set warehouse_id stays constant for a given terminal
set w_id [ RandomNumber 1 $w_id_input ]
set sql2 "select max(d_id) from district"
set d_id_input [ standsql $curn1 $sql2 ]
set stock_level_d_id [ RandomNumber 1 $d_id_input ]
set sql3 "BEGIN DBMS_RANDOM.initialize (val => TO_NUMBER(TO_CHAR(SYSDATE,'MMSS'))
* (USERENV('SESSIONID') - TRUNC(USERENV('SESSIONID'),-5))); END;"
oraparse $curn1 $sql3
if {[catch {oraplexec $curn1 $sql3} message]} {
error "Failed to initialise DBMS_RANDOM $message have you run catoctk.sql as
sys?" }
oraclose $curn1
puts "Processing $total_iterations transactions with output suppressed..."
set abchk 1; set abchk_mx 1024; set hi_t [ expr {pow([ lindex [ time {if { [
tsv::get application abort ] } { break }} ] 0 ],2)}]
for {set it 0} {$it < $total_iterations} {incr it} {
if { [expr {$it % $abchk}] eq 0 } { if { [ time {if { [ tsv::get application
abort ] } { break }} ] > $hi_t } { set abchk [ expr {min(($abchk * 2),
$abchk_mx)}}; set hi_t [ expr {$hi_t * 2} ] } }

```

```

set choice [ RandomNumber 1 23 ]
if { $choice <= 10 } {
if { $KEYANDTHINK } { keytime 18 }
neword $curn_no $w_id $w_id_input $RAISEERROR
if { $KEYANDTHINK } { thinktime 12 }
} elseif { $choice <= 20 } {
if { $KEYANDTHINK } { keytime 3 }
payment $curn_py $w_id $w_id_input $RAISEERROR
if { $KEYANDTHINK } { thinktime 12 }
} elseif { $choice <= 21 } {
if { $KEYANDTHINK } { keytime 2 }
delivery $curn_dl $w_id $RAISEERROR
if { $KEYANDTHINK } { thinktime 10 }
} elseif { $choice <= 22 } {
if { $KEYANDTHINK } { keytime 2 }
slev $curn_sl $w_id $stock_level_d_id $RAISEERROR
if { $KEYANDTHINK } { thinktime 5 }
} elseif { $choice <= 23 } {
if { $KEYANDTHINK } { keytime 2 }
ostat $curn_os $w_id $RAISEERROR
if { $KEYANDTHINK } { thinktime 5 }
}
}
oraclose $curn_no
oraclose $curn_py
oraclose $curn_dl
oraclose $curn_sl
oraclose $curn_os
oralogoff $lda
}
}

```

ABOUT PRINCIPLED TECHNOLOGIES



Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.
